

Para saber más: Excepciones padrones

En el video, usamos la excepción *IllegalStateException*, que es parte de la biblioteca de Java e indica que un objeto tiene un estado invalido. Es posible que haya visto otras excepciones famosas, como *NullPointerException*. Ambos son parte de las excepciones padrones del mundo Java que el desarrollador debe conocer.

Hay otra excepción padrón importante que podríamos haber utilizado en nuestro proyecto. Para ponerlo en contexto, ¿tiene sentido crear una cuenta con una agencia que tiene un valor negativo? Por ejemplo:

```
Cuenta c = new CuentaCorriente (-111, 222); //¿tiene sentido?
```

[COPIA EL CÓDIGO](#)

No tiene ningún sentido. Entonces, podríamos verificar los valores en el constructor de la clase. Si el valor es incorrecto, lanzamos una excepción. ¿Cual? La *IllegalArgumentException*, que es parte de las excepciones de la biblioteca de Java:

```
public abstract class Cuenta {  
  
    //atributos omitidos  
  
    public Cuenta(int agencia, int numero){  
  
        if(agencia < 1) {  
            throw new IllegalArgumentException("Agencia inválida")  
        }  
    }  
}
```

```
    }

    if(numero < 1) {
        throw new IllegalArgumentException("Número de cuen
    }

    //resto del constructor fue omitido
}
}
```

[COPIA EL CÓDIGO](#)

IllegalArgumentException e *IllegalStateException* son dos excepciones importantes que el desarrollador de Java debería utilizar. En general, cuando tenga sentido, utilice una excepción estándar en lugar de crear la suya propia.

¿Todo bien hasta ahora?